

## **Virtual Field Programmable Gate Array Triple Modular Redundant Cell Design**

**Barbara Marty**

**Schafer  
2309 Renard Place SE, Suite 300  
Albuquerque, NM 87106-4267**

**James C. Lyke**

**AFRL/VSSE  
3550 Aberdeen Ave SE  
Kirtland AFB, NM 87117-5776**

**28 April 2004**

**Final Report**

**APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.**



**AIR FORCE RESEARCH LABORATORY  
Space Vehicles Directorate  
3550 Aberdeen Ave SE  
AIR FORCE MATERIEL COMMAND  
KIRTLAND AIR FORCE BASE, NM 87117-5776**

DTIC COPY

AFRL-VS-PS-TR-2004-1093

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data, does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report has been reviewed by the Public Affairs Office and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nationals.

If you change your address, wish to be removed from this mailing list, or your organization no longer employs the addressee, please notify AFRL/ VSSE, 3550 Aberdeen Avenue SE, Kirtland AFB, NM 87117-5776.

So not return copies of this report unless contractual obligations or notice on a specific document requires its return.

This report has been approved for publication.

//signed//

JAMES LYKE  
Project Manager

//signed//

KIRT S. MOSER, DR-IV  
Chief, Spacecraft Technology Division

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. <b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b>					
1. REPORT DATE (DD-MM-YYYY) 28-04-2004		2. REPORT TYPE Final Report		3. DATES COVERED (From - To) 17 Jul 02 - 30 Jan 04	
4. TITLE AND SUBTITLE Virtual Field Programmable Gate Array Triple Modular Redundant Cell Design				5a. CONTRACT NUMBER F29601-98-D-0036 TO 0007	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER 63401F	
6. AUTHOR(S) Barbara Marty; *James C. Lyke				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Schafer 2309 Renard Place SE Suite 300 Albuquerque, NM 87106-4267				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) *AIR FORCE RESEARCH LABORATORY/ VSSE 3550 ABERDEEN AVE. SE KIRTLAND AFB, NM 87117-5776				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/ VSSE	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-VS-PS-TR-2004-1093	
12. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Space systems are becoming increasingly dependent on field programmable gate arrays (FPGAs) and yet the higher gate-count devices are vulnerable to the space environment. To compensate for this effect, triple modular redundancy tools have been developed. Unfortunately these tools complicate the user's design and they cannot cope with the so-called single event functional interrupt (SEFI). In this report, a method referred to as the virtual FPGA has been developed to provide an alternative. It is based on the combination of three FPGAs and radiation-hardened support circuitry. The basic idea involves voting each of up to 512 user input/output pins on all three commercial FPGAs using a radiation-hardened voting circuit. A hardened microcontroller manages the configurations of each of the three FPGAs and monitors upset statistics. Advanced packaging is used to form a component that seems similar in form to the single FPGAs that the concept intends to replace.					
15. SUBJECT TERMS Space Vehicles; FPGA; Space Protection; SEU Mitigation; Multichip Modules; Radiation Testing					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT UNCLASSIFIED	b. ABSTRACT UNCLASSIFIED	c. THIS PAGE UNCLASSIFIED			JAMES LYKE
			Unlimited	32	19b. TELEPHONE NUMBER (include area code) 505-846-5812

Standard Form 298 (Rev. 8-98)  
Prescribed by ANSI Std. Z39.18

**THIS PAGE INTENTIONALLY LEFT BLANK.**

## TABLE OF CONTENTS

<b><u>1.0</u></b>	<b><u>INTRODUCTION</u></b>	<b>.....</b>
<u>1.1.</u>	<u>DESCRIPTION</u>	6
<u>1.2.</u>	<u>THE SIMPLE THEORY OF THE VFPGA</u>	7
<u>1.3.</u>	<u>EVOLUTION OF THE PROJECT</u>	8
<u>1.4.</u>	<u>MODES AND LIMITATIONS OF THE VFPGA BASED ON THE GANGLION DESIGN</u>	9
<u>1.5.</u>	<u>STATUS OF THE VFPGA PROJECT</u>	10
<u>1.5.1.</u>	<u>Summary</u>	11
<b><u>2.0</u></b>	<b><u>TMR CELL DESIGN</u></b>	<b>.....</b>
<u>2.1.</u>	<u>DEFINITIONS</u>	13
<u>2.2.</u>	<u>ACRONYMS</u>	14
<u>2.3.</u>	<u>I/O DESCRIPTIONS</u>	14
<u>2.3.1.</u>	<u>Table of I/O's</u>	14
<u>2.3.2.</u>	<u>Description of I/O's</u>	15
<u>2.4.</u>	<u>OPERATION</u>	23
<u>2.5.</u>	<u>TIMING DIAGRAMS</u>	25
<u>2.6.</u>	<u>CONSIDERATIONS IN TMR CELL / GANGLION DESIGN</u>	27
<b><u>3.0</u></b>	<b><u>CONCLUSIONS / STATUS</u></b>	<b>.....</b>

## LIST OF FIGURES

Figure 1. The Virtual FPGA Concept.....	6
<a href="#">Figure 2. Simplistic comparison of different FPGA hardening approaches.</a> .....	7
<a href="#">Figure 3. VFPGA as a MCM (floorplan).</a> .....	11
<a href="#">Figure 4. Cell Block Diagram</a> .....	12
<a href="#">Figure 5. Tri-state I/O Data</a> .....	13
<a href="#">Figure 6. CRC error count data bits.</a> .....	19
<a href="#">Figure 7. Count bits for Global count type.</a> .....	20
<a href="#">Figure 8. Count bits for Region count type.</a> .....	20
<a href="#">Figure 9. Count bits for Pin count type.</a> .....	21
<a href="#">Figure 10. Output Data select multiplexer.</a> .....	22
<a href="#">Figure 11. Mode Decision Flow Chart</a> .....	28

## 1.0 Introduction

It is fast becoming clear that configurable electronics will be vital in future space systems. Earlier implementations of radiation-hardened field programmable gate arrays (FPGAs), even with their relatively primitive integration scales (measured in single-digit K gates), have become a part of every satellite baseline since their introduction in the mid-late 1990s. It is small wonder then that space users covet the far more powerful FPGAs that have been introduced in the terrestrial market. These devices, with vast integration scales (measured in millions of gates), offer the ability to repetitively reconfigure high-performance electronics in system in response to demanding missions. Unfortunately, the disconnect in gate count between commercial (8M) and rad-hard (0.03M) is a severe barrier. The “straight-up fabrication” of FPGAs in rad-hard form has been a non-starter due to industry concerns of intellectual property compromise. Even with full cooperation, the wholesale conversion of a commercial FPGA will likely require tens of millions of dollars, take many years, and will lock in time a snapshot of a device that may well be obsolete even as the intended end system users complete system builds prior to launch.

The virtual FPGA offers a new cross-cutting strategy to address this problem through the use of selective redundancy, process/design hardening, and aggressive management / containment of observable errors. The approach is “re-useable”, meaning that the methodology is directly applicable to almost any FPGA meeting certain criteria (e.g., no latch-up), side-stepping the “snapshot-in-time” lock-in effect of a large-scale device conversion effort, which has historically been shown as relatively ineffective.

The potential impacts / benefits of the virtual FPGA are powerful and significant. First, VPGA permits rapid implementation of complex, high-performance designs. While in no way a total replacement for custom ASICs, the VPGA, at 3M-8M gates, will permit direct implementation of entire functional block diagrams in a single device, without the ~\$1M NRE and 9 month cost / schedule for every rad-hard ASIC. It is estimated by the industry that high-performance FPGAs can implement 80% of ALL digital designs. Hence time/cost recovery potential is very high. Second, the VPGA permits design change. ASICs must be re-created. A single design flaw can devastate a program if it requires refabrication. FPGA schemes such as VPGA offer the possibility of recovery in < 1 day, not six months. Designs can even be upgraded and design flaws rectified AFTER launch, a clear impossibility for ASICs. Third, the VPGA permits the use of existing commercial design tools. Fourth, the VPGA can combat design obsolescent, even among competing schemes. Fifth, the VPGA is complementary to other hardening efforts, and it can be used to attain additional improvements in reliability/fault management.

This report documents detailed technical analysis and design of components of a concept referred to as the “virtually-hardened field programmable gate array” (VFPGA). The objective of the research effort is to create a compact coarse-grain redundancy management system in which three FPGA devices are voted, pin-for-pin, at the component level. The application-specific integrated circuit (ASIC) responsible for laundering the pins is referred to as the “ganglion” ASIC, so named because of the anticipated > 2,000 pins that would be implemented to support voting 512 user input / output (I/O) pins (including power and ground termini). The ganglion is upset-immune

by design, and as such acts as a firewall to a user design when a single glitch occurs in any FPGA, even a catastrophic glitch, sometimes referred to as a single-event functional interrupt (SEFI). The ganglion itself is managed by a radiation-hardened microcontroller, which contains the bitstreams of the FPGAs in a configuration memory. Upon powerup, the microcontroller retrieves two types of configuration files, the first being a pin function map (for the ganglion), and the second being the configuration memory bitstream of the FPGAs being protected.

The VFPGA effort is implemented over several AFRL contract programs. This report concentrates on the concept of the ganglion ASIC and development of intellectual property suitable for synthesis in a design-hardened ASIC technology.

### 1.1. Description

A simplified representation of the VFPGA is shown in Figure 1. The usage concept for the VFPGA is to present a user with what appear to be an FPGA, but in actuality is a set of components

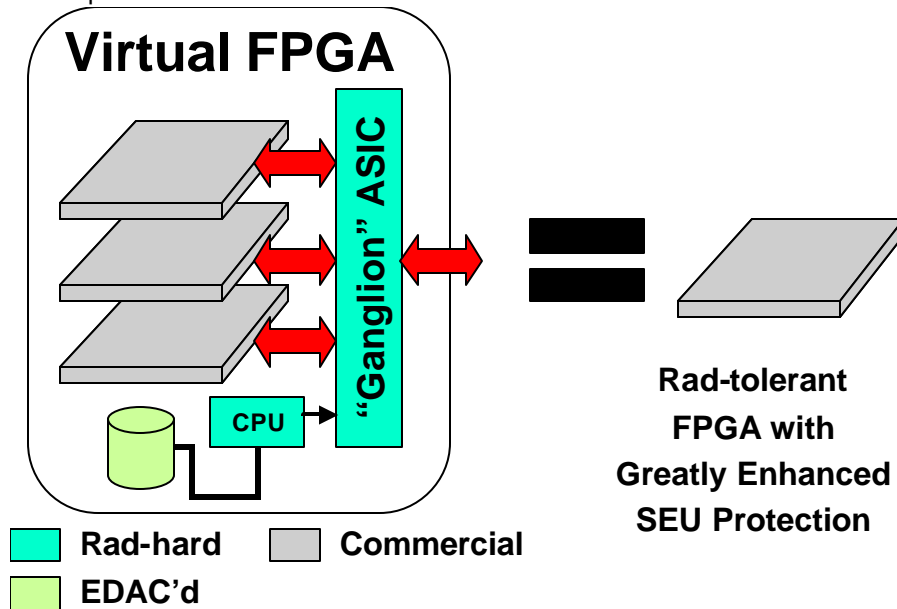


Figure 1. The Virtual FPGA Concept.

**Ganglion ASIC.** The ganglion ASIC is a pin-voting system in which identical user pins from three different FPGAs are combined into a single, protected user pin. The initial plan is to form an array of 512 such cells. The directionality of each pin must be programmed, and this information must be extracted from the FPGA target design before the bitstream is generated. The expected mechanism for this would be a hopefully automatic tool inserted into a standard FPGA synthesis, place-and-route, and bitstream generation tool chain. The other difference in the standard FPGA design and programming flow is the bitstream download. Since two binary structures must be downloaded into the configuration management processor, it is necessary to replace the

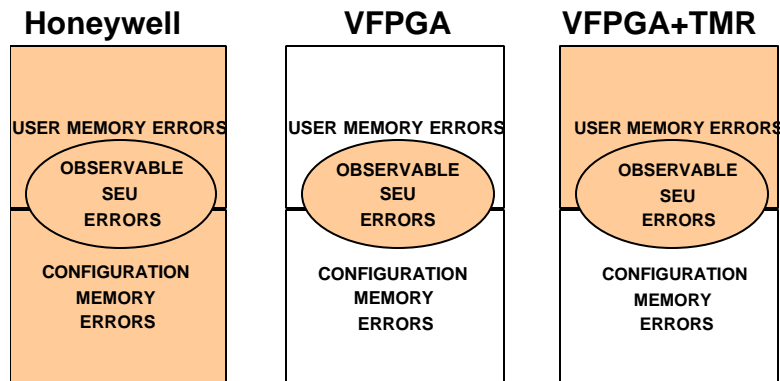


normal FPGA download with a special form that transmits first the user I/O configuration and then the unchanged.

## 1.2. The Simple Theory of the VFPGA

One way to explain very simply the VFPGA concept is to use a comparison of different potential hardening approaches, as shown in Figure 2. In SRAM-based FPGAs, it is necessary to consider configuration and user memory (i.e., state-storage structures). In a user design, the flip-flops and memory cells that form finite state machines (FSMs), registers, and other structures comprise the user memory. On the other hand, the user-inaccessible state storage structures that configure the FPGA to implement the logic, memory, and interconnect needed in user designs comprise the configuration memory. Disrupt the user memory and, for example, a counter may flip to a different count state. Disrupt the configuration memory and the counter may not be a counter any more. In typical designs, most configuration memory is considered benign, meaning that the effective SEU cross section is less than the sum of the area of all configuration memory in general. However, it is desirable to protect all memory structures from the effects of disruption.

The best way to do this is to “brute force” hardened the entire FPGA. The largest complexity device that this has yet been attempted on is a 40K gate equivalent Atmel device, through a NASA program to Honeywell. This creates a bullet-proof FPGA, and requires no thought from the user as to how to introduce mitigation. Unfortunately, the gate count is so anemic when compared to modern FPGAs that, when combined with the expense of the components and their power consumption (not to mention the potentially withering support of the tool chain) the potential uses of this device are very limited.



**Figure 2. Simplistic comparison of different FPGA hardening approaches.**

The VFPGA approach (Figure 2, center) focuses on presenting the user a consistent picture at the boundary of multiple devices. In other words, the VFPGA concentrates on observable errors. The premise in effect is that only observable errors produce errors, regardless of how many upsets are affecting individual devices. As such, VFPGA provides a potentially effective solution for arbitrarily large FPGA devices, including the more complex devices with built in DSP and processor cores, such as the Altera Excalibur or Xilinx Virtex II Pro families. The VFPGA guards against even the

catastrophic failure of a single FPGA, but is susceptible to any single event effect on the remaining two devices.

An additional level of robustness can be created by combining triplication schemes with VFPGA (Figure 2, right). In this case, the user logic can provide an increased interval of protection from cumulative error effects. Obviously, the configuration memory is still vulnerable however, and SEFI problems are not improved in this scheme.

Xilinx recently introduced the XTMR tool, which employs a configuration memory scrubbing mechanism, dramatically increasing the coverage of errors over that shown in the right portion of Figure 2. In this case, only SEFI mechanisms are not addressed with some level of redundancy, but protection from at least one SEFI is guaranteed with the VFPGA concept.

### **1.3. Evolution of the Project**

The VFPGA project evolved from earlier investigations concerning the radiation hardness of the Altera Stratix family of FPGAs. The Stratix FPGA family in many ways is similar to the Xilinx Virtex family. However, Altera introduced as interesting feature potentially useful in identifying single event errors. Contained within the configuration circuitry of Stratix FPGAs is a cyclic redundancy check (CRC) generator, used to verify integrity of the bitstream. Although the primary intended use of this CRC generator is to ensure the initial transfer is not corrupted, the CRC generator can be regenerated on demand while the FPGA is an operation. This mechanism serves as an upset indicator, which can be used to trigger a reconfiguration cycle which essentially resets the configuration of the FPGA to a known condition.

The process of replenishing the configuration of an FPGA during operation for SEU recovery is referred to as "scrubbing." In more standard usage, "scrubbing" in memory systems is used to refresh memory locations during operation to prevent the accumulation of many single bit errors that would otherwise overwhelm the ability of the error detection and correction (EDAC). For example, if an EDAC system is capable of correcting a single bit error in each word of the memory component, the goal of scrubbing is to read and write back each memory word well before two errors can accumulate at any memory location. So long as this can be done successfully, a user never perceives the impact of any errors in the memory system. By contrast, if an FPGA receives a single bit error upset, its impacts can affect the component immediately. In this case, scrubbing might repair the component only after it is too late in a critical application. Since many space users began to worry about this problem, a significant amount of work has gone into creating methodologies and tools to increase robustness of a user design to single bit errors. Synplicity's Synplify tool, for example, provides triple modular redundancy (TMR) support for synthesis involving certain Actel FPGA components. Xilinx and Sandia National Laboratories worked to develop a TMR Tool, which is capable of replicating an entire user design from input to output. Regardless of the effectiveness of these approaches, they introduce considerable overhead to a user design. In fact, the support for this method combined with scrubbing introduces significant burden and overhead to a potential user. Furthermore, the effectiveness of these approaches are limited by the small but finite possibility of SEFIs.

Consideration of this problem led to the idea of creating a virtual FPGA. The foundation of the virtual FPGA concept was based on the notion that a user should be able to largely pursue design without repetitively inventing solutions to the SEU problem. As a

currently stands, SEU mitigation and complex FPGAs is ad hoc, and significant nonrecurring engineering (NRE) is expended across the Aerospace community in dealing with the problem. If, on the other hand, a virtual FPGA could be created and sold as a modular solution, then it is no longer necessary to continuously reinvent the solution approach. Furthermore, through the use of advanced packaging and architecture techniques, the insertion of a virtual FPGA need not represent a 3X size penalty, owing to the eventual plans to implement it within the floor plan of a compact multichip module.

At this point (early 2003), the Schafer effort focused more directly on the problems associated with creating a virtual FPGA. For the most part, this was done without invalidating the previous work to create a test board for the Stratix radiation test. Schafer developed a plan building on the previous work, in which one of the test FPGA devices would be used to implement the functionality of the ganglion, and the exposed (device-under-test) FPGA would be partitioned into three pseudo FPGAs.

Schafer looked toward the creation of the ganglion ASIC IP, and examined areas ancillary problem areas, such as the Advanced Instrument Controller (AIC). The AIC was originally developed as an integrated microcontroller for the Mars Deep Space Two (DS2) mission. Since DS2, the core design had undergone significant refinement and was being contemplated as the control management processor for the ganglion ASIC in the virtual FPGA design. Some of the Schafer effort was expended on examining the feasibility of creating another AIC. This effort was eventually tabled due to a complementary effort at Mission Research Corporation to create a single chip AIC.

#### ***1.4. Modes and Limitations of the VFPGA Based on the Ganglion Design***

The ganglion ASIC in its current design embodiment provides a number of interesting features. First, it provides an en masse statistics base on single event effects at different levels of granularity, from single pins (for each of 512) to upsets on the entire component. This unprecedented level of visibility makes possible the development of mission specific reconfiguration policies, based on the slope of upsets in devices. For power savings, it is possible to power-down two of the three devices. This is of particular benefit to low-earth orbiting missions in which most of the orbital profile is benign. The VFPGA can take advantage of built-in CRC information or ignore it, as the user may see fit.

The VFPGA operation can be viewed as a stop light. Without upsets, the VFPGA is in a "green" state. Upon the first upset, the VFPGA is potentially susceptible to a second hit, but is still functioning correctly. This is the "yellow" state. Other errors may accumulate and be rectified effectively by the en masse TMR. On the other hand, certain designs and error modes may spread rapidly owing to a "cone of influence" propagating from a single fault. When two or more hits affect the same regions of more than one FPGA, the operation of the VFPGA is not reliable, which corresponds to the "red" state.

The concept of operation of the VFPGA is then best summarized as operating without concern until "yellow". Once "yellow", the system in which the VFPGA is located should strive to perform a total reset of the VPGA as soon as possible. This is distinctly different from other mitigation schemes in which "green" can go to "red" with no "grace period". The duration of the "grace period" (i.e., the "yellow-to-red" interval) is purely a function of circumstance, the orbital profile, the method of design.

## Design enhancements to the Ganglion

A number of other intriguing possibilities exist to make the Ganglion ASIC even more useful.

1. One is rapid-context switching. In this case, the control CPU could trigger the VFPGA to select only one of three distinct personalizations, which could be alternated within a clock cycle. This amounts to a rapid-fire context switch, with a maximum of three contexts.
2. The VFPGA could be expanded from three to four or more different FPGAs, thereby further improved robustness.
3. The phase edges of individual devices could be skewed to provide some robustness to single event transient
4. In dose rate environments, the VFPGA could be overwhelmed by rail span collapse in multiple devices. Under these conditions, no mitigation strategy can be effected, and complex FPGAs become unreliable altogether. If the ganglion and control management processor are built in a dose-rate hardened CMOS technology, however, it is possible to create a firewall concept. In the firewall concept, a per-pin safehold level is defined, and upon a trigger signal, the ganglion moves all pins electrically to the safehold level. This is a potentially useful mechanism for circumvention and recovery. Until a special "clear" signal is asserted (upon reconfiguration of all devices within the VFPGA), the safehold levels are maintained, thus providing an orderly scheme for bringing disrupted devices online.

### **1.5. Status of the VFPGA Project**

The VFPGA project has received support from the Air Force Research Laboratory / VSSE and from Missile Defense Agency (MDA). While the effort in this contract produced the essential basis of the ganglion ASIC, much work is still needed, and at the time of this writing is being pursued in follow-on and concurrent contract and in-house work.

The planned eventual embodiment of the VFPGA is in a ball grid array multichip module (MCM) package as motivated by the Figure 3 depiction. This MCM would be formed using the high density interconnect (HDI) patterned overlay process, which has been successfully used to implement a number of complex MCM designs for aerospace applications, ranging from jet fighters to interplanetary spacecraft. The challenges in doing this include: (1) wiring supply of several high pincount components; (2) obtaining functional parts in bare die form. If successful, the VFPGA can be formed as a very compact package, preferably a package that is "isomorphic" to that used in a standard FPGA. It is important to note that this particular embodiment places each FPGA device in a tiled arrangement. Stacked arrangements could be done, but such configurations increase the likelihood of a pathological strike in which an energetic particle penetrates the entire stackup. The horizontal tiling, while not totally immune to this effect, considerably reduces the cross section or probability of its occurrence.

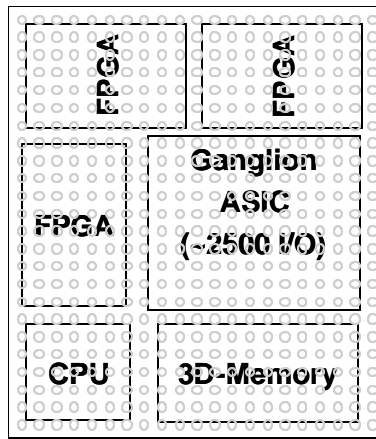


Figure 3. VFPGA as a MCM (floorplan).

### 1.5.1. Summary

The VFPGA is a scheme for systematic TMR at an entire-component level. It relies on the use of a radiation-hardened voting ASIC to manage redundancy of observable errors on a pin-by-pin basis. Advanced packaging is proposed as a way of managing the very large number of input/output pins, so that a VFPGA is not much larger than a single FPGA component.

The remainder of this document addresses the Ganglion ASIC design. The next chapter describes the TMR cell array, which amounts to the heart of the Ganglion.

## 2.0 TMR Cell Design

The cell is considered a hardened by design core module. It takes data from 3 data sources (FPGA's), corrects the data, and outputs the corrected data. It counts the number of errors detected from each data source independently and keeps a record of the error counts. When requested, the cell will transmit the error counts on the **sd1**, **sd2**, and **sd3** serial data lines. It can also count CRC errors if the data sources have CRC capability and output the count on the **sdcrcl** serial data line. The cell can work in conjunction with a microprocessor that makes decisions based on the error counts of the data sources. Figure 1 shows a simplified block diagram of the cell.

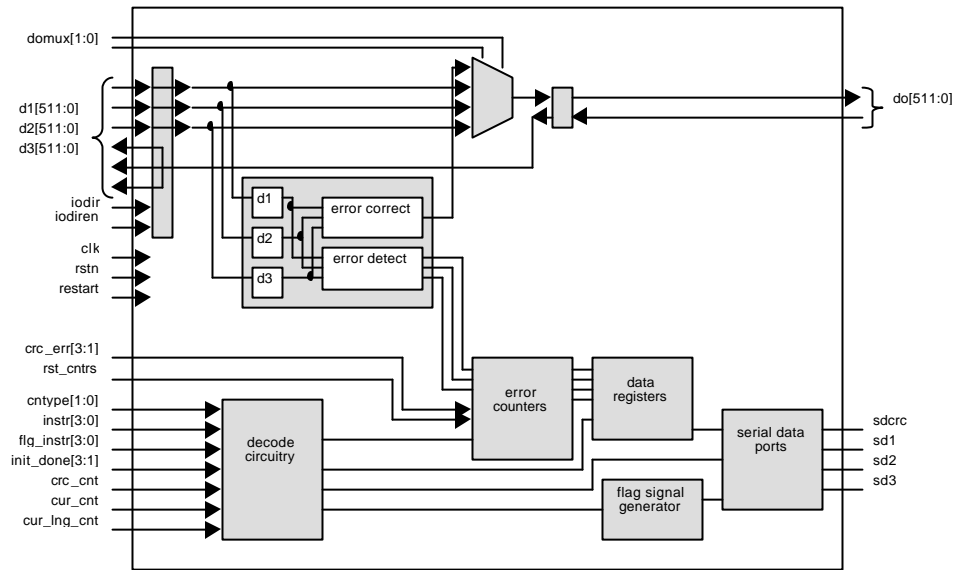


Figure 4. Cell Block Diagram

The data I/O's are bidirectional, so the data can also come from the outside world into the cell. If the data passes through the cell from the outside world, the cell disperses the data to the three data sources. A simplified diagram of the tri-state I/O data is shown in Figure 2. If data travels from the outside world to the 3 data sources, there will be no EDAC or error counting on those particular data lines. The data direction is set following a Power on Reset (**rstn**) or a **restart** command. The **dir** input signal is a serial stream of 512 bits of data where each bit 0-511 pertains to the I/O pin number and is used to set the direction enable for the data I/O pins. The **diren** signal is the gate for the I/O direction enable serial stream. It is high when the data direction bits are being shifted into the cell and goes low when the last bit is shifted in. A timing diagram of the two signals is shown in Waveform 1 of the Timing Diagrams section in this data sheet. Each of the 512 bits in the serial stream will set 4 I/O pins direction. Three of the 4 are the direction enables for I/O's between the data sources (FPGA's) 1, 2, and 3, and the cell.



## 2.2. Acronyms

CRC	Cyclic Redundancy Check
EDAC	Error Detection and Correction
TMR	Triple Modular Redundancy
FPGA	Field Programmable Gate Array
ASIC	Application Specific Integrated Circuit
IC	Integrated Circuit
I/O	Input/Output
LSB	Least Significant Bit
MSB	Most Significant Bit

## 2.3. I/O Descriptions

This section gives a table of all the inputs and outputs of the TMR Cell. It gives details on what each input does and explains the outputs as well.

### 2.3.1. Table of I/O's

clk	input	Global clock.
rstn	input	power on reset.
restart	input	Restarts the system to an initial state.
indn	input	Initialization (or configuration) of data sources is complete.
crcerr[3:1]	inputs	CRC error signals from data sources (FPGA's). These inputs must be externally grounded if unused.
diren	input	Gate for dir serial input data stream. This signal is high when the dir data stream that sets the tri-state I/O's to input's or output's is being shifted in. It is low otherwise.
dir	input	Serial input data stream that is 512 bits long for stipulating data direction on the tri-state I/O's. 0 = input on the d1, d2, and d3 data lines and output on the do data lines. 1 = output on the d1, d2, and d3 data lines and input on the do data lines.
cntyp[1:0]	inputs	Count type select bits. 00 = No counting. 01 = Global error count mode. Increment count each positive clock edge by total number of data mismatches for each FPGA. 10 = Region error count mode. Increment each regional counter every positive clock edge by total number if data mismatches in each region for each FPGA. 11 = Pin error count mode. Increment each data pin counter every positive clock edge for each FPGA.



instr[3:0]	inputs	Instruction bits select the terminal count and the terminal count bit width. See tables 1, 2, and 3.
finstr[3:0]	inputs	Flag instruction bits select whether or not to set a flag when terminal bit count has been reached and what the terminal count number will be. See tables 1, 2, and 3.
crccnt	input	Gate instructing the cell to transmit the CRC count on the sdrcrc serial data line.
curcnt	input	Gate instructing the cell to transmit the terminal count bits only sd1, sd2, and sd3 serial data lines.
curlcnt	input	Gate instructing the cell to transmit the all the error count bits on the sd1, sd2, and sd3 serial data lines.
rstcnt	input	Counter reset instructs the counters that have not yet reached terminal count to reset.
domux[1:0]	inputs	Data output multiplexer selects which data to output on the do lines that have been set to outputs. 00 = Output the triple voted data from the 3 data sources (FPGA's). 01 = Output the data from data source 1 (FPGA 1). 10 = Output the data from data source 2 (FPGA 2). 11 = Output the data from data source 3 (FPGA 3).
sdrcrc	output	Serial data CRC error count. Bits 0-3 are the crc error count from data source 1. Bits 4-7 are the crc error count from data source 2. Bits 8-11 are the crc error count from data source 3.
sd1	output	Serial data error count from FPGA 1.
sd2	output	Serial data error count from FPGA 2.
sd3	output	Serial data error count from FPGA 3.
d1[511:0]	I/O	Birdirectional data to/from data source 1. Any unused pins should be externally grounded.
d2[511:0]	I/O	Birdirectional data to/from data source 2. Any unused pins should be externally grounded.
d3[511:0]	I/O	Birdirectional data to/from data source 3. Any unused pins should be externally grounded.
do[511:0]	I/O	Birdirectional data to/from "outside world". Any unused pins should be externally grounded.

### 2.3.2. Description of I/O's

#### 2.3.2.1. System Clock – clk

This is the system global clock.

#### 2.3.2.2. System Reset – rstn

This is the system power on reset.

#### **2.3.2.3. Restart:**

This input resets all counters and initializes all internal registers back to their original reset value of zero. It is used as a reset or initialization state after reconfiguration of one or more data sources. If one of the data sources is taken offline, the system needs to be synchronized back to an initial state.

#### **2.3.2.4. Initialization (or Configuration) done - indn:**

This input signal goes from low to high when the 3 data sources are done configuring and initializing. It should remain high unless a power on reset or restart command is issued.

#### **2.3.2.5. CRC error - crcerr:**

These input signals come from the three FPGA's. They toggle "high" when CRC errors are detected in the configuration memory of the FPGA's. The cell counts these separately from the data errors. If unused, these pins need to be tied to ground externally.

#### **2.3.2.6. Gate for I/O tri-state data direction enable - diren:**

This signal is the gate for the serial data direction stream. When it is high, the input data on the dir line is serially transmitted into the cell. It needs to remain high until each data direction bit has shifted into the cell. When this signal goes low again, the i/o data direction is registered and remains until a power on reset or a restart command occurs. A timing diagram of the I/O data direction select is shown in Waveform 1 of the Timing Diagram section.

#### **2.3.2.7. I/O Data Direction Serial Stream – dir:**

This is the serial data bit stream that sets the tri-state I/O buffers to inputs or outputs. Once the direction is set, the data becomes available on the tri-state I/O pads as soon as the internal sysen signal goes high, which is the second rising edge of the clock after the diren signal goes low as shown in Waveform 1 of the Timing Diagrams section.

On the tri-states that are set as inputs ("0"), the data comes from the 3 data sources, through the cell to the outside world. The data on these lines gets EDAC and any errors are counted and stored. On the tri-states that are set as outputs ("1"), the data comes from the outside world, passes through the cell, and on to the 3 data sources. EDAC and Error counting on these lines does not occur.

### 2.3.2.8. Count type - cntyp[1:0]:

These two inputs select which type of error counting to use. The count type can only be changed following a power on reset (rstn) or a restart (restart) command and after the I/O direction enable cycle. The count type command should be issued directly after the I/O direction enable gate goes low.

The count type instructions are:

- 00 = No counting.
- 01 = Global counting. All data input pins with errors get counted globally each clock cycle. It counts total errors for each FPGA each clock cycle. Each FPGA has its own independent counter, so there are 3 separate counters. This type of counting uses the least amount of internal switching, but only narrows down the FPGA with the most errors.
- 10 = Region counting. The error counters are grouped into 32 regions of 16 data input pins. This narrows down the regions where problems are occurring. Each error within the group of 16 gets counted every clock cycle. This type of counting uses more counters than the global count, so switches more, but narrows down into smaller regions of each FPGA, where more errors are occurring.
- 11 = Pin counting. There is an error counter for each input data pin individually. This type of counting uses the most counters and therefore the most power, but narrows down errors to each individual data input pin on each FPGA.

A timing diagram showing of the count type, instruction, and flag instruction is shown in Waveform 2 of the Timing Diagrams section of this data sheet.

### 2.3.2.9. Instruction – instr[3:0]

The instruction tells the cell what the maximum error counts should be. The terminal count bit width is the overflow bit. It is set when the maximum instructed count is reached. The instruction set is shown in Table 1. For Global count type there is only one terminal count bit for each FPGA counter. For Region count type there are 32 terminal count bits for each FPGA. For Pin count type there are 512 terminal count bits for each FPGA. The instruction should be issued directly after the gate for the I/O direction enable goes low.

**Table 1. Instruction Set.**

Instr [3:0]	Terminal Count Bit Width	Terminal Count
0000	3	4
0001	4	8
0010	5	16
0011	6	32

0100	7	64
0101	8	128
0110	9	256
0111	10	512
1000	11	1,024
1001	12	2,048
1010	13	4,096
1011	14	8,192
1100	15	16,384
1101	16	32,768
1110	17	65,536
1111	18	131,072

A timing diagram showing the instruction is shown in Waveform 2 of the Timing Diagrams section of this data sheet.

#### **2.3.2.10. Flag Instruction – finstr[3:0]:**

The flag instruction, if used, tells the cell when to set an alert signal if the selected number of terminal bits has been reached. After a power on reset or restart command and I/O direction enable cycle, the flag instruction should be issued. If the flag instruction is set to 0000, the flag signal is turned off and there will be no warning signal from the cell that error counts are reaching terminal counts. The flag instruction must be less than or equal to the terminal count number or it will not get set. If there is no request to send data on the serial data lines and the stipulated number of terminal count bits has been reached, the serial data line from the corresponding FPGA will go high and remain high until either a power on reset, restart, or request for error data (curcnt or curlcnt) occurs. The flag instruction used with each count type is shown in Table 2. The flag will get set when the number of terminal count bits stipulated in the table is reached.

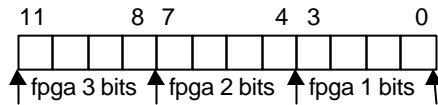
**Table 2. Flag Instruction Set.**

finstr [3:0]	Flag with Global cntyp	Flag with Region cntyp	Flag with Pin cntype
0000	0	0	0
0001	1	2	2
0010	1	3	4
0011	1	4	8
0100	1	5	16
0101	1	6	32
0110	1	7	64
0111	1	8	128
1000	1	9	256
1001	1	10	256
1010	1	11	256
1011	1	12	256
1100	1	13	256
1101	1	14	256
1110	1	15	256
1111	1	16	256

A timing diagram showing the flag instruction is shown in Waveform 2 of the Timing Diagrams section of this data sheet.

#### 2.3.2.11. CRC Error Count request – crccnt:

When this signal goes high, the cell transmits the current CRC error count on the serial data **sdcr** line. It transmits 12 bits of data as shown in Figure 3. The first 4 bits ([3:0]) are the CRC error count from FPGA 1. The second 4 bits ([7:4]) are the CRC error count from FPGA 2. The third 4 bits ([11:8]) are the CRC error count from FPGA 3.

**Figure 6. CRC error count data bits.**

The timing diagram for the CRC error serial data bit stream is shown in Waveform 4 of the Timing Diagrams section.

#### 2.3.2.12. Current Count Request – curent:

When this input goes high, the cell transmits only the terminal count error bits corresponding to FPGA's 1, 2, and 3, on the serial data lines, **sd1**, **sd2**, and **sd3**, respectively. This input must remain high until the final bit has shifted out. If this input remains high after all the bits have shifted out, zeros will be shifted out. If this input goes low before the end of transmission, the cell will stop transmitting data. If this input toggles low and then high again, the new terminal count data bits will start transmitting. A timing diagram of the serial data bit stream when the **curcnt** command is issued is shown in Waveform 4 of the Timing Diagrams section.

If the count type is Global, only one bit will be transmitted and it will be a "1" if the terminal count has been reached and a "0" otherwise. Figure 4 shows a diagram of the count bits. The "n" above the bold box represents the terminal count bit, which was selected by the flag instruction.

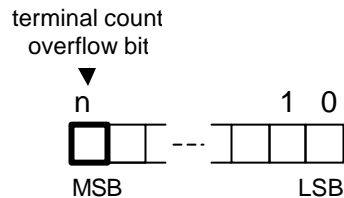


Figure 7. Count bits for Global count type.

If the count type is Region, a maximum of 16 transmitted bits will have terminal count data. Only the bits that correspond to regions that have reached terminal error count will be set. Figure 5 shows a diagram of the count bits. The "n" above the bold boxes represents the terminal count bit, which was selected by the instruction, for each of the 32 regions.

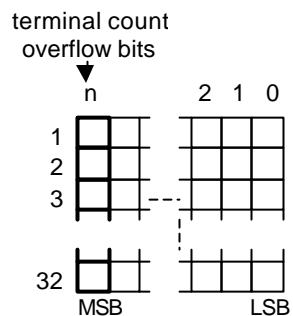


Figure 8. Count bits for Region count type.

If the count type is Pin, 512 bits will be transmitted. Only the bits that correspond to input data pins that have reached terminal error count will be set. Figure 6 shows a diagram of the count bits. The "n" above the bold boxes represents the terminal count bit, which was selected by the instruction, for each of the 512 I/O pins.

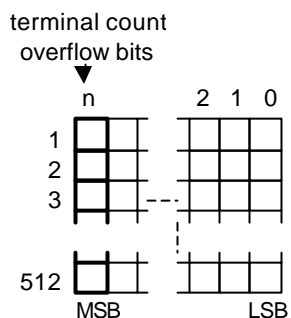


Figure 9. Count bits for Pin count type.

### 2.3.2.13. Current Long Count Request – curlcnt:

When this input goes high, the cell transmits the current count for each data input pin. The bit order will be I/O pin 0, bit 0 first, bit 1 next, and so on until the count bit width for pin 0 is reached and then I/O pin 1 bit 0, bit 1, and so on until all the count bits corresponding to each FPGA, 1, 2, and 3, on the serial data lines, **sd1**, **sd2**, and **sd3**, respectively have been transmitted. This input must remain high until the final bit has shifted out. If this input remains high after all the bits have shifted out, zeros will be shifted out. If this input goes low before the end of transmission, the cell will stop transmitting data. If this input toggles low and then high again, the new count data will start transmitting. The timing diagram showing the serial data bit stream when this command is issued is shown in Waveform 4 of the Timing Diagrams section.

If the count type is Global, the number of data bits shifted out is determined by the instruction bit width. For example, if the terminal count bit width is set to 8, 8 bits will be shifted out where the 8<sup>th</sup> bit is the terminal count bit. Refer to Figure 4 for the diagram of the count bits. The least significant bit (LSB) is shifted out first, and the most significant bit (MSB), which is the terminal count bit, is shifted out last.

If the count type is Region, the number of data bits shifted out is determined by the instruction bit width times 32. For example, if the terminal count bit width is set to 8, 8 bits will be shifted out for each of the 32 regions for a total of 256 bits on each of the **sd1**, **sd2**, and **sd3** lines corresponding to those FPGA's. The 8<sup>th</sup> bit from each region's error count is the terminal count bit for that region. Refer to Figure 5 for the diagram of the count bits. The LSB is shifted out first, and the MSB, which is the terminal count bit, is shifted out last from each region.

If the count type is Pin, the number of data bits shifted out is determined by the instruction bit width times 512. For example, if the terminal count bit width is set to 8, 8 bits will be shifted out for each of the 512 I/O pins for a total of 4096 bits on each of the **sd1**, **sd2**, and **sd3** lines corresponding to those FPGA's. The 8<sup>th</sup> bit from each I/O pin error count is the terminal count bit for that I/O pin. Refer to Figure 6 for the diagram of

the count bits. The LSB is shifted out first, and the most significant bit MSB, which is the terminal count bit, is shifted out last from each I/O pin.

#### **2.3.2.14. Reset Counters – rstcnt:**

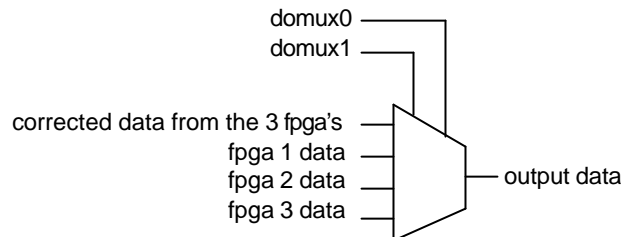
This input will reset the counters that have not yet reached terminal count. It will not reset the counters that have reached terminal count. If the counters are not reset periodically and are allowed to continue counting, over an extended period of time, the counters could reach terminal count due to periodic inadvertent bit flips. If this happens, the system could get reconfigured and restarted needlessly.

#### **2.3.2.15. Data Output Multiplexer – domux[1:0]:**

These two inputs are the select lines for the data output multiplexers. They should be set to 00 when the system is in normal operation after a power on reset or restart. When they are set to 00, the output is the error corrected data from the 3 data sources. If these signals get set to something other than 00, the output comes directly from the data source selected and no error correction is done on the output data.

- 00 = corrected data from the 3 data sources.
- 01 = data from FPGA 1 only.
- 10 = data from FPGA 2 only.
- 11 = data from FPGA 3 only.

If domux is selected to be something other than 00, error counting stops. Figure 7 shows a diagram of the data output mux. If one of the other data lines besides 00 is selected, this is an indication that an external element (microprocessor) has made a decision to reconfigure one of the FPGA's due to the number of errors it was experiencing.



**Figure 10. Output Data select multiplexer.**

#### **2.3.2.16. Serial Data CRC Error Count – sdcr:**

This is the serial data output for the CRC error count for each of the three FPGA's. It is a twelve bit serial stream with data output as shown in Figure 3. The timing diagram for



the serial stream is shown in Waveform 4 in the timing Diagrams section. The data is serially output as explained in section 4.2.11 above.

#### 2.3.2.17. Serial Data lines 1, 2 and 3 – sd1, sd2, and sd3:

This is the serial data output for the error count corresponding to FPGA's 1, 2 and 3, respectively. The timing diagram for the serial data streams is shown in Waveform 4 in the Timing Diagram section. The data is serially output as explained in sections 4.2.12, and 4.2.13 above.

#### 2.3.2.18. Data Lines – d1[511:0], d2[511:0], and d3[511:0]:

These are the input/output data lines between data sources, FPGA's 1, 2, and 3, and the cell. All unused pins should be externally grounded. After power up or a restart command all the data I/O's get initialized to be either an input or output. The data input on the I/O pins that are set to inputs gets corrected, if errors are detected, and output on the corresponding output data line.

#### 2.3.2.19. Data Lines – do[511:0]:

These are the input/output data lines between the cell and the outside world. All unused pins should be externally grounded. After power up or a restart command all the data I/O's get initialized to be either an input or output. The data lines from section 4.2.18 selected to be inputs will be outputs on these lines. The data lines from section 4.2.18 selected to be outputs will be inputs on these lines.

### 2.4. Operation

Following a power on reset or restart command the cell is in an idle state. All internal registers are set to zero. All tri-state I/O's are inputs until the data direction has been set with the **diren** gate signal and **dir** serial data direction stream. Once configuration and initialization of the system is complete and the I/O data direction has been set, the cell is ready to receive the count type and instruction sets for operation. Once the cell receives the count type and both sets of instructions, it will begin counting. The count type, instruction, and flag instruction should all be sent simultaneously.

Once the count type has changed from 00, error counting on the 3 data lines will start and the new count type will be registered. The count type can only be changed to a different type of counting after a power on reset or a restart command. Table 3 shows the count type bit selects. The count types are explained further in section 4.2.8 above.

**Table 3. Count type bits**

<b>cntyp</b>	<b>Description</b>
00	No counting

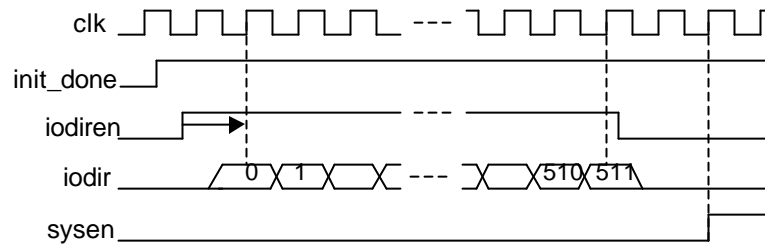
01	Global type counting
10	Region type counting
11	Pin type counting

If any of the **crcnt**, **curcnt**, or **curlcnt** signals are high, which are when the current error data is being requested and the data is transmitting on the serial data lines, and these lines are toggled low and then high again, the new error data will begin transmitting. The **crcnt** signal is independent of the **curcnt** and **curlcnt** signals. Only one of the two **curcnt** or **curlcnt** signals should be high at a time. If one of them is high and the other goes high before the first goes low, any new error data will begin transmitting and the old data will be lost. If both of these signals are low and the output data on one of the serial data lines is high, it means the cells internal terminal count error limit has been reached for that FPGA and the flag is set.

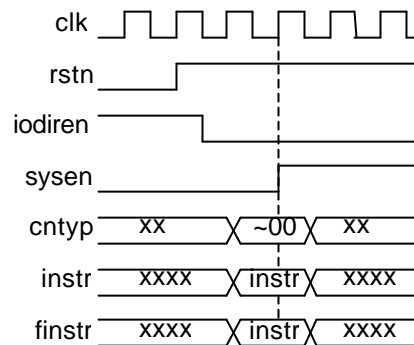
The serial data output lines operate independently of the rest of the system, so once a request is made for the error count data, it gets stored in the serial data shift registers and begins shifting. The I/O data lines continue to operate normally and error counting continues as well.

While the error counters are counting, if any of the counters reach terminal count, that particular counter will stop counting. This feature keeps the cell from unnecessary switching and saves power. If the reset counters command is issued, the counters that have reached terminal count will not get cleared. These counters will only get cleared with a power on reset or restart command.

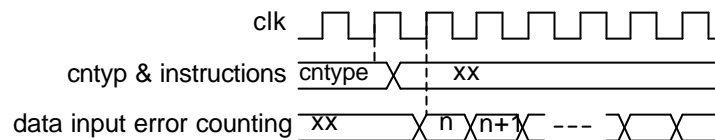
## 2.5. Timing Diagrams



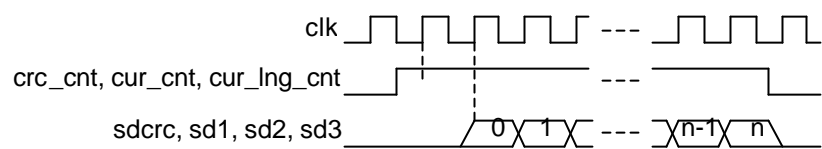
**Waveform 1. I/O Data Direction Selection Timing Diagram**



**Waveform 2. Setting Count Type and Instructions Timing Diagram**



**Waveform 3. Input Data from 3 Data Sources Timing Diagram**



**Waveform 4. Timing Diagram for CRC Error data Request and Serial Data Bits**

## **2.6. Considerations in TMR Cell / Ganglion Design:**

The decision making process that can be used for determining if an FPGA should be reprogrammed does not exist within this cell design. If this cell is used in conjunction with a microprocessor, the microprocessor will determine the parameters for reprogramming. If the microprocessor is reprogrammable, the parameters can be changed. Some considerations to look at:

- are the majority of the bit errors coming from the same data source?
  - if so, are the errors coming from the same pins or region?
- are CRC errors being looked at and if so, are they coming from the same data source?
- have the counters been reset?

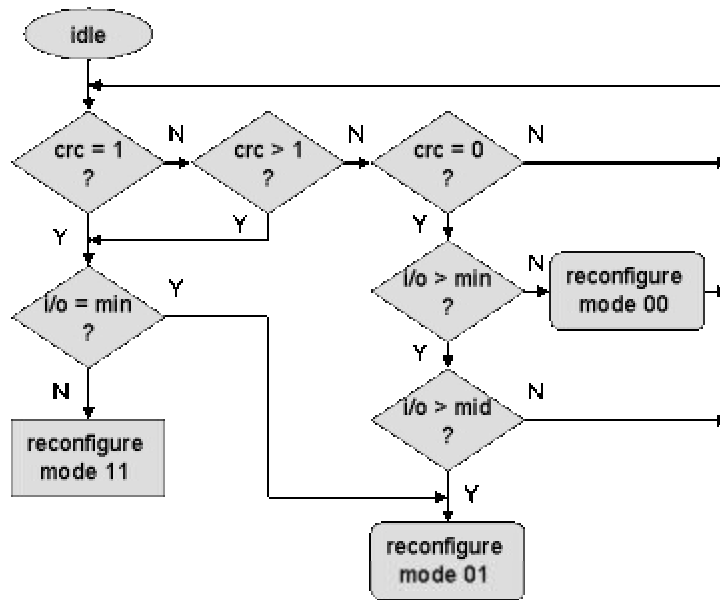
Three modes should be looked at in the decision making process. The first, mode 00, is non-immediate. Non-immediate means that attention is not required immediately. The errors are typically bit flips and, while they get counted as errors from the data source, immediate attention is not required.

The second, mode 01, is semi-immediate. This mode means that attention is required, but the data source can remain in operation until a reconfigure can be done. The errors are continuous and are occurring frequently enough that the data source needs to be reconfigured, but are not severe enough to take the data source off line immediately. The data source can wait until a more convenient time frame occurs where it can get reconfigured.

The third, mode 11, is immediate. Immediate attention is required on one of the data sources and it should be taken off line and reconfigured immediately. The data source should not be used until after it has been reconfigured and a restart command has been issued to initialize the system.

When the third mode of operation has been established based on the errors the cell has detected and counted, and one of the three data sources is taken off line, one of the remaining two data sources will provide the data to the outside world. Since the cell stores the detected errors, this data can be used to determine which of the remaining two sources has been more stable. Whichever source has been determined more stable can be used for the data output. Once all three sources are operating normally again, the output data will once again be the triple voted data.

The process used to determine if one of the FPGA's should be reprogrammed depends on the parameters set for the frequency and type of errors occurring. Having CRC checking greatly improves the significance of the problems occurring since it looks at the configuration memory of the FPGA's. An example flow chart for the decision making process is shown in Figure 8.



**Figure 11. Mode Decision Flow Chart**

The chart shows CRC error considerations and I/O error considerations. If the CRC errors are 0, there are no problems detected with configuration memory and this is considered minimum (min). If the CRC errors are greater than one, there are problems within configuration memory, but because the crc error pin is toggling, the errors in configuration memory are sporadic. This is considered middle range (mid). If the CRC errors are stuck at one, configuration memory has a more severe problem. This is considered maximum (max). The same follows with the I/O errors detected. A few errors are within a minimum range, more errors are within a middle range, and a lot of errors have reached a maximum range.

The percentage of total errors to pins, regions, or entire FPGA is the parameter that should be able to be reset or reprogrammed. For example, if counting pin I/O errors for each FPGA and one of the FPGA's has reached terminal count on half of its I/O pins, there is a maximum percentage of I/O errors. If the CRC error count on this same FPGA is one or greater, there is also a problem with the configuration memory. This FPGA should be taken off line immediately since it shows severe damage.

If one of the FPGA's is taken off line, the next consideration is which of the remaining FPGA's to use for the data output. Since the data is stored in the cell, it can be looked at to determine which of the remaining two FPGA's has been more stable and that FPGA's data will be used for the output to the outside world.

### **3.0 Conclusions / Status**

This report has described the virtual FPGA (VFPGA) concept and logic design of the key TMR management application-specific integrated circuit (ASIC). Chapter 1 provided an overview of the VFPGA concept, and Chapter 2 detailed the datasheet-level description of the ganglion ASIC or TMR cell array.

At the time of this writing, AFRL seeks funding to carry out an implementation of the basic concept as a design-hardened ASIC and multi-chip module (MCM). In the meantime, preparation is underway to test a scaled version of the VFPGA using two FPGAs. One FPGA is mounted in a fixture to permit exposure to heavy ions for single event upset (SEU) testing. This FPGA is partitioned into three pseudo-FPGAs, representing in simulation the three separate FPGA devices that would be used in the actual VFPGA implementation. The second FPGA is a service FPGA that will carry an implementation of the Ganglion/TMR cell array. It is shielded from exposure, mimicking in effect a rad-hard part. This setup will be implemented as part of a more comprehensive FPGA test to study SEU phenomena in modern FPGAs, and it will establish viability of the overall VFPGA concept.

## DISTRIBUTION LIST

DTIC/OCP 8725 John J. Kingman Rd, Suite 0944 Ft Belvoir, VA 22060-6218	1 cy
AFRL/VSIL Kirtland AFB, NM 87117-5776	1 cy
AFRL/VSIH Kirtland AFB, NM 87117-5776	1 cy
Schafer 2309 Renard Place SE, Suite 300 Albuquerque, NM 87106-4267	1 cy
Official Record Copy AFRL/VSSE/James Lyke	1 cy